

## **REMARKS/ARGUMENTS**

### **Double Patenting Objection**

The Examiner objected to claims 1 and 4 under 37 CFR 1.75 as being a duplicate of claim 6. Independent claim 1 has been amended to include the limitation "wherein at least one of the state value lists indicates a pattern for at least two of the plurality of counters", which overcomes this objection.

### **35 U.S.C. § 103(a) Rejections**

The Examiner rejected claims 1-3 and 5 under 35 U.S.C. §103 over U.S. Patent No. 7,072,880 (Beesley) in view of U.S. Patent Application Publication No. 2004/0098549 (Dorst). In response, the Applicant has amended these claims to define with particularity that a state value list indicates a value/pattern value for at least two of the counters.

The claimed invention is directed to multi-counter evaluation. According to the claims, multi-counter evaluation is performed using a merged finite-state machine that represents the finite-state machines of the individual counters of the multi-counter. The merged finite-state machine is augmented with state value lists instead of state values. Each state value list indicates which counter scores receive which values for the corresponding state, wherein at least one of the state value lists indicates a value/pattern for at least two of the plurality of counters.

The merged finite state machine recited in claims 1-3 and 5 requires two individual finite-state machines sharing a common state in the merged finite-state machine. Looking at the specification (page 6, second table, and Figure 1), Figure 1 shows states 0-4 forming a first finite-state machine. States 0 and states 5-11 in Figure 1 form a second finite-state machine. Both the first and second state machines each have a separate counter (specification at page 6, lines 16-27, and page 7, lines 1-7). In the merged finite-state machine the two machines that have been merged share a common state (state 0) that is the beginning state of both individual finite-state machines. The merged finite-state machine has state value lists that replace the plurality counters (one for the first finite-state machine and one for the second finite-state machine) as required by claims 1-3 and 5.

The Examiner pointed to Figures 13 and 14, col. 13, lines 56-67, and col. 14, lines 16-24 and 30-33 of Beesley as teaching a merged or augmented finite-state machine having at least one state each corresponding to a multiplicity of states each of a different one of the single-

counter finite state machines and augmented with state value lists. The Applicant respectfully disagrees.

Instead, Beesley teaches a finite-state network that contains a series of domains (finite-state machines) 350-354 (Beesley, Figures 8 and 9). Each domain 350-354 has a counter that counts the number states indicating the number of paths leading from the previous state (Beesley, col. 9, lines 32-39). The counts/counters for each domain are separate and unique to each domain (Beesley, col. 4, lines 60-65). A single domain does not have a plurality of counters associated with the domain as required by claims 1-3 and 5. The domains are a series of state machines that are strung together serially (see Figures 8 and 9). The finite state-network corresponds to strings/words, while each state machine corresponds to a domain of sub-strings/formants. Since the finite-state network is a concatenation of finite-state machines, the finite-state network does not have at least one state each corresponding to a multiplicity of states each of a different one of the finite-state machines as required in claims 1-3 and 5.

Compare this to Figure 1 from the application where the finite state machines in the merged finite state machine have a common state (state 0). It could be argued that domain 350 in Beesley Figures 8 and 9 shares a common state (the state represented by the number 4 in domain 350 in Figure 9) and therefore reads on claims 1-3 and 5. However, domain 350 only has a single counter (Beesley, col. 9, lines 32-39, Figure 9), not a state value list representing a plurality of counters as required in claims 1-3 and 5. Beesley sets forth steps for the preparation and use of a network for substring-to-number mapping and for retrieving related information, and not what the Examiner purports.

In order for Beesley to disclose what is in claims 1-3 and 5, two of the domains 350-354 would have to be merged. For example, if the node designated by 4 in domain 350 of Beesley Figure 9 was merged with either one of the nodes designated with a 2 in domain 351 of Beesley Figure 9 to form a single common state as shown for state 0 in Figure 1 of the application, then Beesley would arguably disclose a merged finite-state machine as claimed in claims 1-3 and 5. However, this is not the case.

Likewise, Dorst fails to teach two finite-state machines sharing a common state in a merged finite-state machine. Instead, Dorst is directed at managing memory devices in computer systems.

Furthermore, Beesley and Dorst fail to teach state value lists where each state value list indicates which counter of the multi-counter receives which value for the state of the merged finite-state machine, wherein **at least one of the state value lists indicates a value for at least two of the plurality of counters** (see specification Figure 1, state 9 value list shows

counter 1 being updated by a value of 2.0 and counter 2 being updated by a value of 3.0) as required by claim 1.

Beesley discloses that a unique index or an array of labeled indices is returned when one or more substrings are identified, in col. 13, lines 56-67, and col. 14, lines 16-24. The returned index/indices do not indicate which counter will receive which value. In fact, the returned index/indices have nothing to do with a counter, but instead are an index into a database (Beesley, col. 14, lines 3-6).

Beesley does disclose that each domain has a separate counter that counts number states indicating the number of paths leading from the previous state (Beesley, col. 9, lines 32-39, Figure 9). Since each domain only has a single counter, there is no indication of which counter will receive which value or values for at least two counters as required in claim 1. Beesley has no reason to indicate which counter to update or values for at least two counters because each domain only has a single counter and counter information is not shared/updated between domain counters.

Likewise, Dorst fails to teach state value lists where each state value list indicates which counter of the multi-counter receives which value for the state of the merged finite-state machine, wherein **at least one of the state value lists indicates a value for at least two of the plurality of counters** as required by claim 1.

Instead, Dorst teaches three count-down counters that are loaded based on the occurrence of **separate** states (events) in a finite state machine (Dorst, paragraph [0071]). The first counter is loaded based on states ARCS, AWCS, RCPW, WCPW, RWAIT, and WWAIT. The second counter is loaded based on states ARES, AWES, REPW, and WEPW. The third counter counts the number of beats remaining in a CSI transfer and is loaded with a value for the requested number of beats (Dorst, paragraph [0072]). In Dorst, **only one counter is updated per state**. The first counter's six events are different events than the second counter's four events. The third counter is loaded based on a CSI transfer, which is not one of the events used for the first or second counters. For Dorst to read on the newly amended section of claim 1, multiple counters would have to be updated based on a single state, which is not disclosed in Dorst. Therefore, both Beesley and Dorst fail to teach state value lists where each state value list indicates which counter of the multi-counter receives which value for the state of the merged finite-state machine, wherein at least one of the state value lists indicates a value for at least two of the plurality of counters as required by claim 1.

The Applicant therefore requests that the Section 103(a) rejection of claim 1 be withdrawn. Likewise, independent claims 2, 3, and 5 contain limitations similar to independent

claim 1 and therefore should be allowed. Dependent claim 4 depends from independent claim 3 and should be allowed.

The Examiner's objections and rejection having been properly responded to and overcome, the Applicant suggests that the application is now in condition for allowance. The Applicant therefore requests that the application be reconsidered and thereafter be passed to issue.

The Applicant considers the foregoing to be dispositive of all issues in the application. But if the Examiner should deem that a telephone interview would advance the prosecution, he is invited to call the Applicant's attorney at the telephone number listed below.

The Commissioner is hereby authorized to charge any fees that may be required, or to refund any overpayment, to Deposit Account No. 501602.

Respectfully submitted,

**Eric T. Bax**

By: /Douglas M. Grover/

Douglas M. Grover  
Corporate Counsel  
Reg. No. 52,974  
303-538-4926

Date: June 11, 2009

Avaya Inc.  
Docket Administrator  
307 Middletown-Lincroft Road  
Room 1N-391  
Lincroft, NJ 07738